

Validación Cruzada (Cross-Validation)

Justin Brad Rodriguez Sanchez - jusrodriguez@unal.edu.co

Cristian Javier Medina Barrios - crmedinab@unal.edu.co

Estephanie Perez Mira - eperezmi@unal.edu.co

Cristian Camilo Cubillos Reyes - ccubillos@unal.edu.co

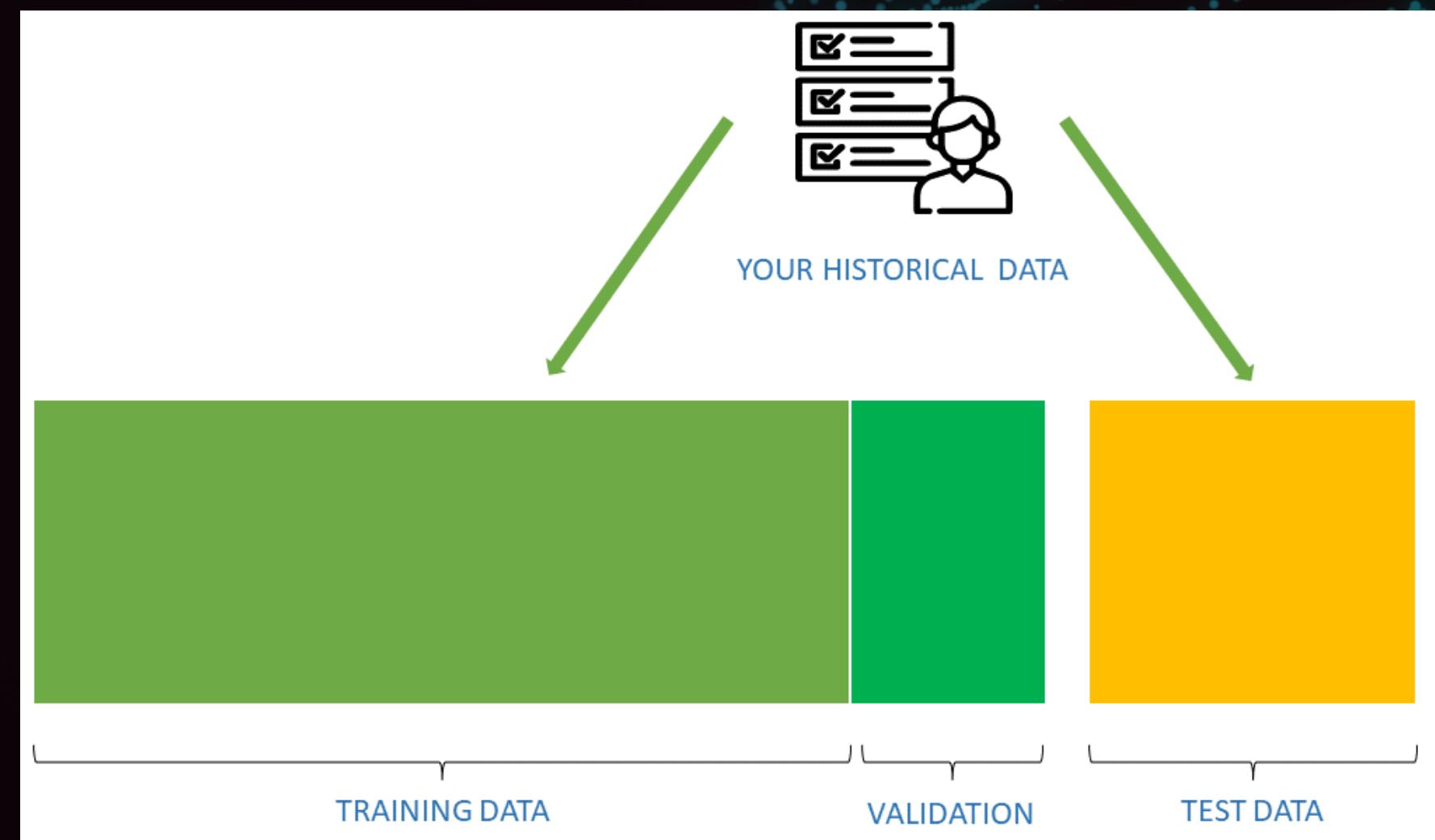
Introducción: el sobreajuste en el entrenamiento de modelos

- Sobreajuste: cuando un modelo se ajusta demasiado a los datos de entrenamiento.
- Resulta en un muy buen rendimiento en datos de entrenamiento, ¡pero falla con datos nuevos!

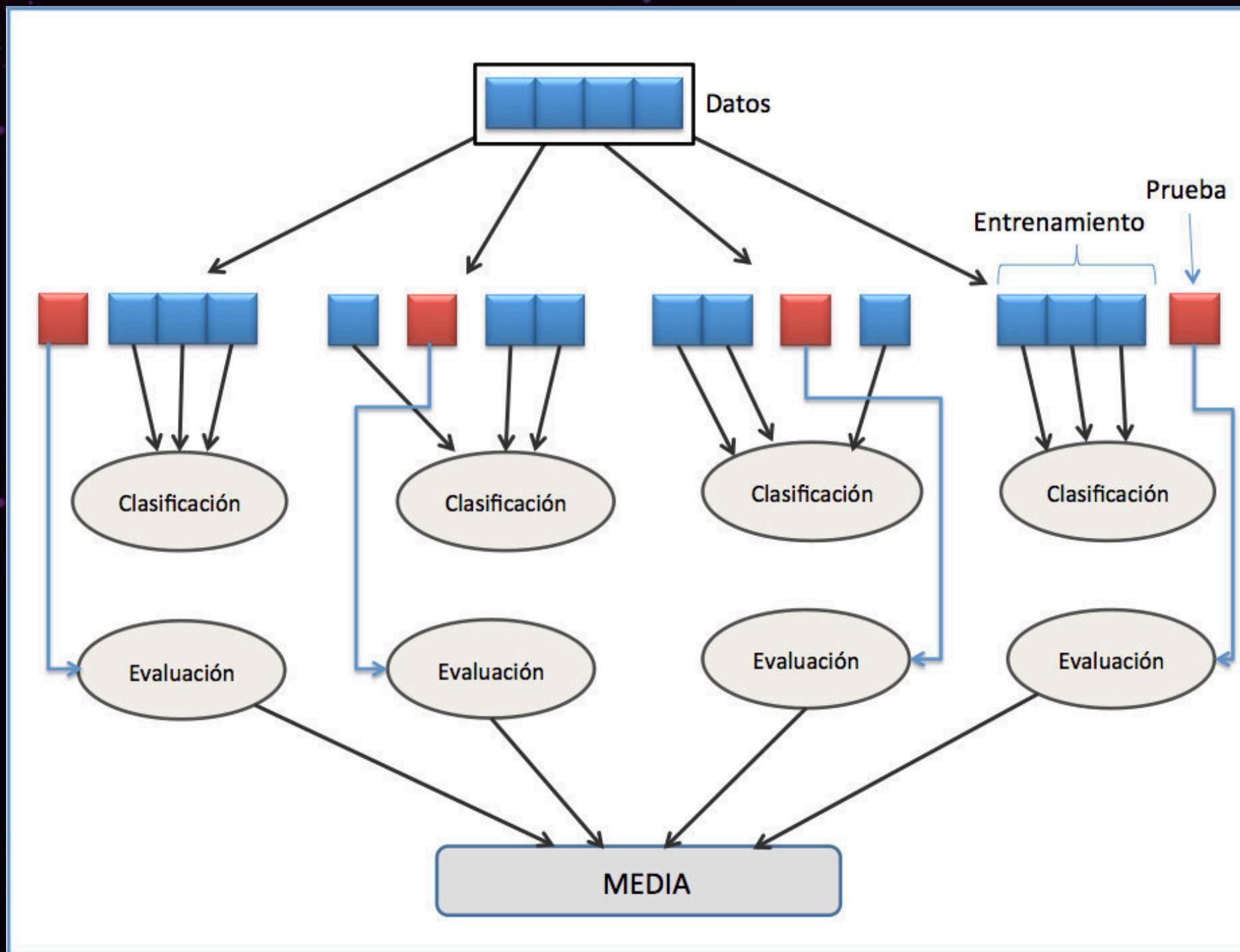
¿Qué podemos hacer para mitigarlo?

Separar el conjunto de datos de entrenamiento en tres conjuntos:

- Entrenamiento
- Validación
- Prueba



¿Qué es la validación cruzada?

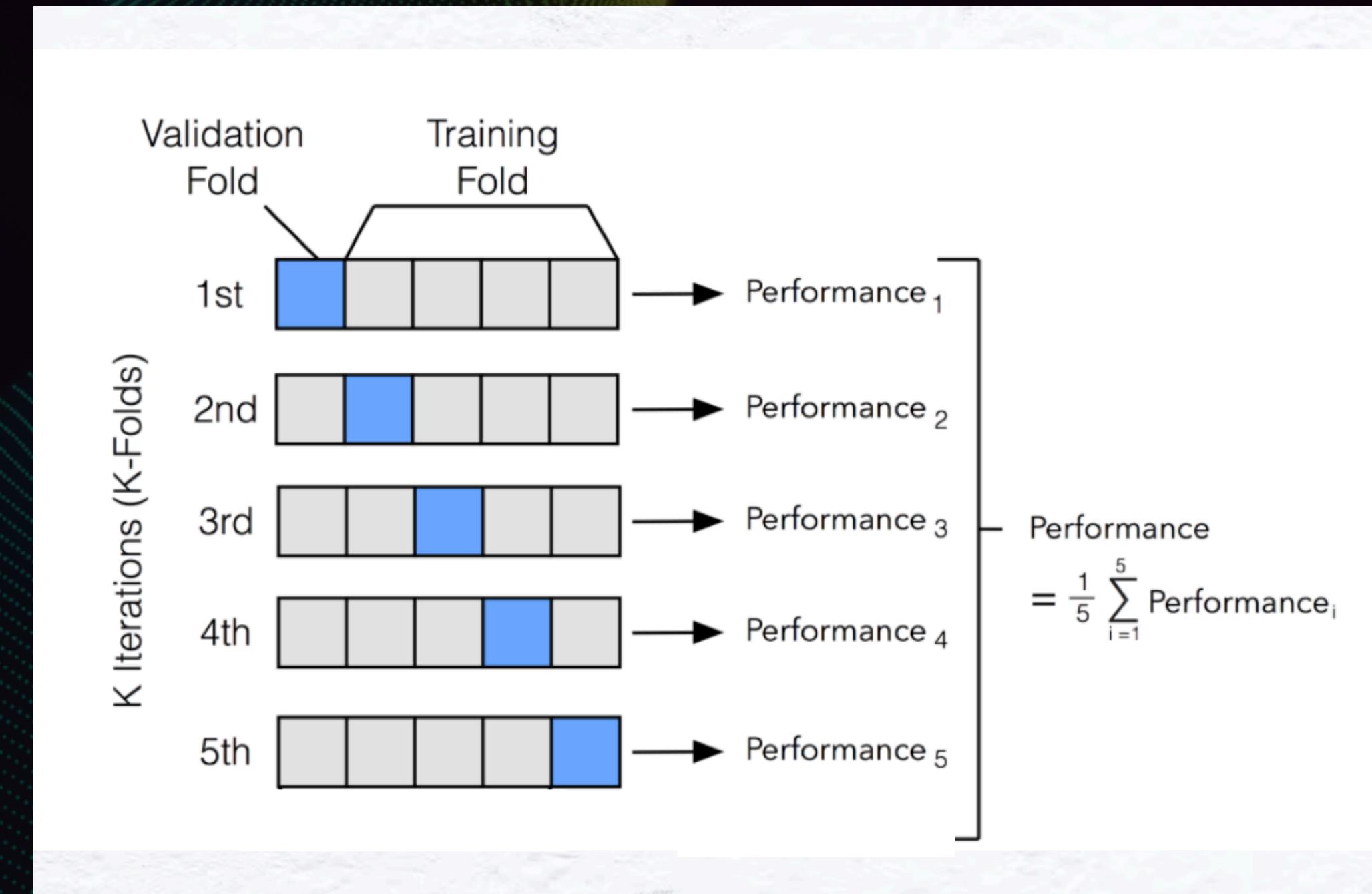


- Es una técnica para evaluar modelos de forma más robusta.
- Divide el conjunto de datos en K subconjuntos (folds).
- Cada fold se usa una vez para validación, mientras los demás se usan para entrenamiento.

Ejemplo: esquema k-fold cross validation, con $k=4$ y un solo clasificador

¿Cómo funciona K-Fold Cross Validation?

- Valores típicos: $K = 5$ o 10 .
- Cada muestra se usa $K-1$ veces para entrenar y 1 vez para validar.
- Métrica final: promedio de los resultados por fold.





Beneficios

- Uso eficiente de todos los datos.
- Menor riesgo de sobreajuste a una sola partición.
- Métricas más estables (ej. accuracy, F1).

Limitaciones y Cuándo Evitarla

Situación	¿Usar K-Fold?	Alternativa
Dataset muy grande	✗	Train/Val/Test único
Dependencia temporal o secuencial	✗	Validación por bloques
Dataset pequeño/moderado	✓	K-Fold o Leave-One-Out

EJEMPLO PRÁCTICO SKLEARN

- Dataset de clasificación con 1797 imágenes.
- Validación Cruzada con $K = 5$ (5 folds).
- Cada vez, el modelo se entrena con 80% y se valida con 20% distintos.
- Métricas como accuracy y F1-score en cada fold.

```
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_digits

digits = load_digits()
x = digits.data # Imágenes (cada una como vector de características)
y = digits.target # Etiquetas (números del 0 al 9)

# Accuracy
accuracy_scores = cross_val_score(RandomForestClassifier(), x, y, cv=5, scoring='accuracy')

# F1 macro (promedia F1 por clase)
f1_scores = cross_val_score(RandomForestClassifier(), x, y, cv=5, scoring='f1_macro')

print("Accuracy por fold:", accuracy_scores)
print("Accuracy promedio:", accuracy_scores.mean())

print("\nF1-score por fold:", f1_scores)
print("F1-score promedio (macro):", f1_scores.mean())
```

RESULTADOS

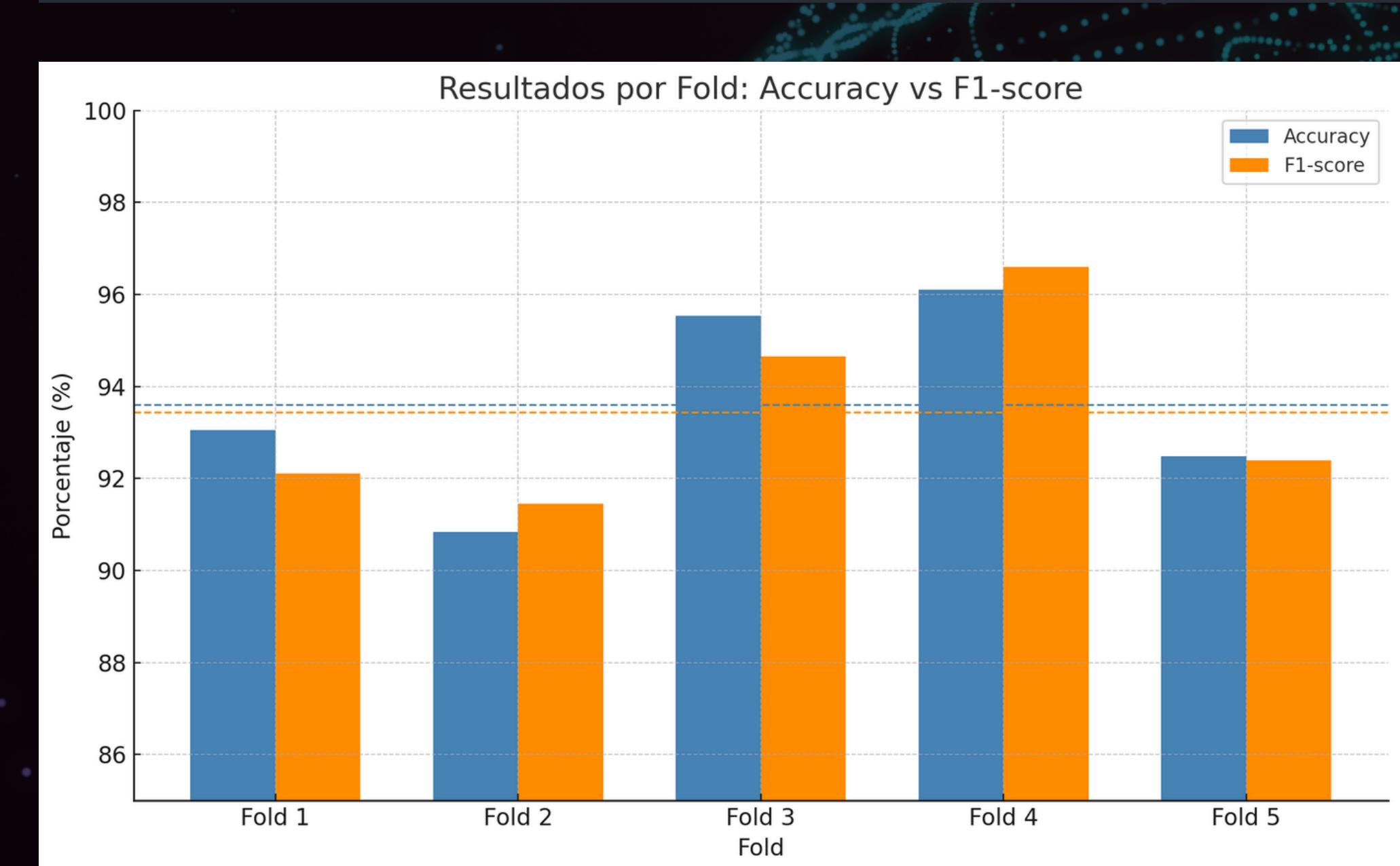
Validación cruzada con 5 folds:

- Accuracy promedio: 93.6 %
- F1-score promedio: 93.4 %
- Resultados consistentes en todos los folds.
- Buen desempeño general y balanceado entre clases.

El modelo predice bien en promedio y también generaliza correctamente en todas las clases (0-9).

```
Accuracy por fold: [0.93055556 0.90833333 0.95543175 0.96100279 0.92479109]
Accuracy promedio: 0.9360229031259673
```

```
F1-score por fold: [0.92109285 0.91458485 0.94653898 0.96602723 0.92390824]
F1-score promedio (macro): 0.934430429495006
```



¿Y EN VISIÓN POR COMPUTADORA COMPLEJA?

Casos como **detección de objetos** o **segmentación**:

- Validación cruzada es posible, pero el proceso es más complejo.
- Se debe dividir cuidadosamente el dataset en varios folds manualmente.
- Muy útil en **investigación con pocos datos** (ej. imágenes médicas).
- Las imágenes y etiquetas deben organizarse en carpetas por fold

Permite **evaluar modelos robustamente**, incluso con poca información.



¿Por qué importa evaluar bien?

- Entrenar ≠ validar
 - No basta con que el modelo funcione durante el entrenamiento. Lo que realmente importa es su desempeño en datos no vistos.
- Métricas sin contexto pueden engañar
 - Una sola métrica no cuenta la historia completa. Validar bien significa hacerlo con varias particiones y con múltiples indicadores.
- El promedio no lo es todo
 - También se deben observar la variabilidad y la consistencia entre folds.



Generalización y Robustez

- Generalización ≠ Memorizar
 - Un modelo puede tener alta precisión en entrenamiento pero fallar en datos nuevos
- Robustez del modelo
 - Si el desempeño es consistente en diferentes particiones, el modelo es más robusto y confiable.
- Rigor Experimental
 - En IA y visión por computador, es fundamental demostrar resultados que se mantengan fuera del set de entrenamiento.



Gracias

